DBAR: An Efficient Routing Algorithm to Support Multiple Concurrent Applications in Networks-on-Chip

Sheng Ma*
School of Computer
National University of Defense Technology
Changsha, China
masheng@nudt.edu.cn

Natalie Enright Jerger
Department of Electrical and Computer Engineering
University of Toronto
Toronto, Canada
enright@eecg.toronto.edu

Zhiying Wang
School of Computer
National University of Defense Technology
Changsha, China
zywang@nudt.edu.cn

ABSTRACT

With the emergence of many-core architectures, it is quite likely that multiple applications will run concurrently on a system. Existing locally and globally adaptive routing algorithms largely overlook issues associated with workload consolidation. The shortsightedness of locally adaptive routing algorithms limits performance due to poor network congestion avoidance. Globally adaptive routing algorithms attack this issue by introducing a congestion propagation network to obtain network status information beyond neighboring nodes. However, they may suffer from intra- and inter-application interference during output port selection for consolidated workloads, coupling the behavior of otherwise independent applications and negatively affecting performance.

To address these two issues, we propose Destination-Based Adaptive Routing (DBAR). We design a novel low-cost congestion propagation network that leverages both local and non-local network information for more accurate congestion estimates. Thus, DBAR offers effective adaptivity for congestion beyond neighboring nodes. More importantly, by integrating the destination into the selection function, DBAR mitigates intra- and inter-application interference and offers dynamic isolation among regions. Experimental results show that DBAR can offer better performance than the best baseline algorithm for all measured configurations; it is well suited for workload consolidation. The wiring overhead of DBAR is low and DBAR provides improvement in the energy-delay product for medium and high injection rates.

Categories and Subject Descriptors

B.4.3 [**Hardware**]: Input/Output and Data Communications—*Interconnections*; C.1.2 [**Computer Systems Organization**]: Multiple Data Stream Architectures—*Interconnection architectures*

General Terms

Design, Performance

Keywords

Networks-on-chip, Routing Algorithm, Workload Consolidation

*This research was carried out while Sheng Ma was a visiting international student at the University of Toronto supported by a CSC scholarship.

ISCA'11, June 4–8, 2011, San Jose, California, USA.

1. INTRODUCTION

Given the difficulty of extracting parallelism, it is quite likely that more than one application will run concurrently on a many-core system [19, 28, 2], often referred to as workload consolidation. Significant research exists on maintaining isolation and effectively sharing on-chip resources such as caches [40] and memory controllers [31]. The network-on-chip (NoC) [7] is another, less-explored example of a shared resource where one application's communication may degrade the performance of another. This work focuses on improving performance and providing isolation for workload consolidation via the routing algorithm.

For high performance, consolidated workloads place different requirements on the routing algorithms. First, the routing algorithm should provide sufficient adaptivity to avoid network congestion. Second, it should not leverage superfluous information leading to inaccurate estimates of network status. Most importantly, it should provide dynamic isolation among different applications. Existing routing algorithms are unable to meet all these needs. Oblivious routing algorithms, such as DOR, ignore current network status, resulting in poor load balancing across channels. Adaptive routing algorithms offer the ability to avoid congestion by supporting multiple paths between a source and destination; a selection strategy is applied to choose between multiple outputs. Most existing selection strategies do not offer both adaptivity and isolation.

The selection strategy should choose the channel that will route the packet to its destination along the path with the least congestion. A local selection strategy leverages only local knowledge, which tends to violate the global balance intrinsic to traffic [17]. Neighbors-on-Path (NoP) selection strategy addresses this issue by using the status of nodes adjacent to neighboring nodes [1]. However, this strategy ignores the status of neighboring nodes, and offers sub-optimal performance for fully adaptive routing algorithms.

Globally adaptive routing, such as Regional Congestion Awareness (RCA) [17], utilizes a congestion propagation network to leverage both local and non-local information

to make a choice; however, it introduces excess information when selecting the output port and offers no isolation among different applications, leading to performance degradation for consolidated workloads. As shown in Section 3, this excess information can be classified as intra- and inter-application interference. Interference makes the performance of applications less predictable.

Considering the future prevalence of server consolidation and the need for performance isolation, an efficient routing algorithm should combine high adaptivity with dynamic workload isolation. Therefore, we believe utilizing precise information is optimal; redundant or insufficient information easily leads to inferior performance. Based on this understanding of network flow, we introduce Destination-Based Adaptive Routing (DBAR), a novel adaptive routing algorithm well suited to workload consolidation.

We design a low-cost congestion information propagation network to leverage both local and non-local network status, giving DBAR high adaptivity. Furthermore, DBAR's selection strategy chooses the output port by only considering the nodes that a packet may traverse, while ignoring nodes located outside the minimum quadrant defined by the current location and the destination node. Thus, it eliminates redundant information and can dynamically isolate applications in different regions. By eliminating interference and offering high adaptivity, DBAR outperforms other routing algorithms for all evaluated network configurations.

This paper makes the following primary contributions:

- Analyzes the limitations of other selection strategies including local, NoP and RCA and proposes a novel destination-based selection strategy that affords sufficient adaptivity for network congestion and dynamic isolation among different applications.
- Explores the effects of intra- and inter-application interference and demonstrates that the amount of congestion information considered impacts performance, especially for consolidated workloads.
- Designs a low-cost congestion information propagation network with only 3.125% wiring overhead to leverage both local and non-local network status.

2. BACKGROUND

In this section, we discuss related work in application mapping and adaptive routing algorithms design.

Since the arrival order and execution time of consolidated workloads cannot be known at design time, run time application mapping techniques are needed [21, 5, 4, 26]. Approaches to offline resource allocation for a single application include a branch and bound algorithm [21] and a two-step genetic algorithm [26]. Mapping each application to a near convex region provides the optimal NoC configuration for workload consolidation [5, 4]. Since most application mapping techniques consider the Manhattan distance between the source and destination but not the routing paths [5, 26], our routing algorithm is complementary to these techniques.

As shown in Fig. 1, an adaptive routing algorithm consists of two parts: the routing function and the selection strategy [1]. The routing function computes the set of possible output channels according to the current and destination locations and the selection strategy chooses one of these channels based on some network status information. Many

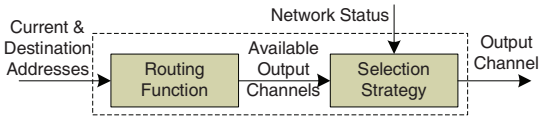


Fig. 1: The structure of an adaptive routing algorithm.

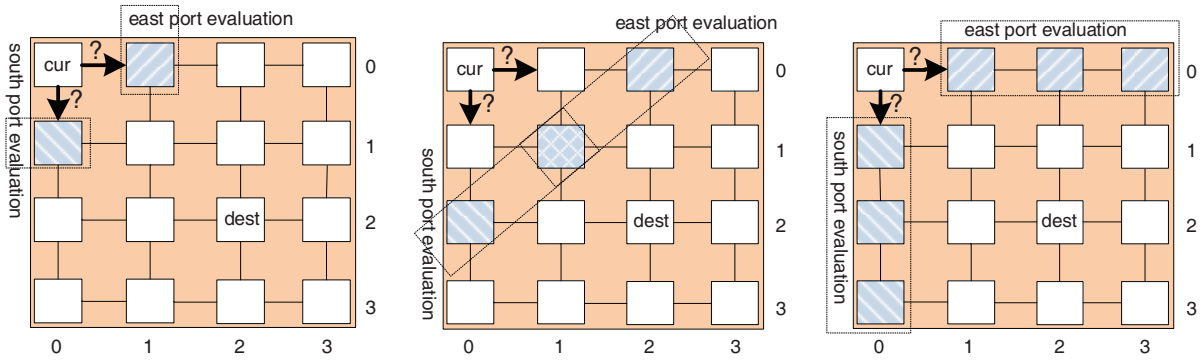
selection strategies have been evaluated including zigzag, XY, no turn, minimum congestion, and maximum flexibility in prior work [16, 35, 14, 29, 9].

A routing function must provide deadlock avoidance [3, 6, 10, 11, 16]. Seminal deadlock-avoidance theories [6, 10, 11] split a physical channel into several virtual channels (VC). Dally and Seitz give the necessary and sufficient condition for designing deadlock-free deterministic routing [6]. Duato further enumerates theories for deadlock-avoidance in adaptive routing [10, 11, 12]. These theories are powerful tools for designing a fully adaptive routing algorithm which can route packets along all minimal paths between the source and destination. Our proposed DBAR achieves deadlock-freedom based on Duato's theory. Turn model routing achieves deadlock avoidance without VC support [16, 3]. Turn model routing algorithms offer partial adaptivity as not all minimal paths between the source and destination are usable.

Off-chip networks are constrained by pin bandwidth, but the abundant wiring resources in NoCs allow easier implementation of congestion propagation mechanisms. Therefore, the NoC paradigm has sparked renewed interest in adaptive routing algorithms. DyAD combines the advantages of both deterministic and adaptive routing schemes [20]. DyXY uses dedicated wires to investigate the status of neighboring routers [27]. A low-latency minimal adaptive routing algorithm performs lookahead routing and pre-selects the optimal output port [24]. The selection strategies of these designs [20, 27, 24] all leverage the status of the neighboring nodes. Instead, Neighbors-on-Path (NoP) makes a selection based on the condition of the nodes adjacent to neighbors [1].

RCA is the first work utilizing both the local and non-local information to improve load balancing in NoCs [17]. However, this algorithm introduces interference in the congestion calculation, especially under workload consolidation. Redundant information may degrade the quality of the congestion estimates; to combat this, Ramanujam and Lin [33] propose a technique to eliminate excess information by integrating the destination into the selection procedure. They maintain per-destination delay estimates in each router, and use these estimates to steer the output selection [33]. They use a dedicated network to sequentially transmit delay information for each network node. However, this mechanism requires long latency for each router to calculate the estimates for all other network nodes. Despite leveraging a similar observation regarding congestion information, our implementation is quite intuitive and each node can obtain timely network status. Furthermore our design considers the performance when running multiple concurrent applications.

BLBDR [34] provides strict isolation between adjacent applications by statically configuring connectivity bits and offers partial adaptivity based on turn restrictions. Moreover, when multiple output ports are available, bLBDR chooses the optimal port based on local information. In contrast, we design a novel selection strategy that offers dynamic isolation between regions and achieves full adaptivity based on



(a) A scenario for LOCAL. (b) A scenario for NoP. (c) An intra-region interference scenario for RCA-1D.

Fig. 2: Packet routing example (the current router is (0,0) and the destination is (2,2)).

Duato's theory. The salient feature of DBAR is that it utilizes both local and non-local information while dynamically isolating different applications in a NoC.

3. MOTIVATION

We motivate the need for a novel routing algorithm from two directions. First, the selection strategy should have enough information about network conditions to offer effective congestion avoidance. Both local and NoP [1] selection strategies lack enough information, leading to sub-optimal performance. Second, RCA [17] utilizes a light-weight monitoring network to obtain global network information; however, its performance suffers from intra- and inter-application interference. DBAR offers a middle ground between these extremes.

3.1 Insufficient information

The local selection strategy (LOCAL) leverages the conditions of neighboring nodes when choosing the output channel. These conditions may be free buffer slots [20, 27, 24, 17, 1], free VCs [9, 17], crossbar demands [17] or a combination [17]. Fig. 2 shows a packet at router (0,0) that needs to be routed to (2,2). Since both the east and south ports are admissible outputs, a selection strategy is required. LOCAL only uses the information about the nearest nodes ((0,1) and (1,0)). Without any information about the condition of the nodes beyond neighboring nodes, it cannot avoid network congestion more than one hop away from current node.

The NoP selection strategy uses the status of nodes adjacent to neighboring nodes as shown in Fig. 2(b). The limitation of the NoP selection strategy is that it ignores the status of neighbors ((0,1) and (1,0)); it makes decisions based only on the conditions of nodes two hops away. In the example, for east output evaluation, it considers nodes (0,2) and (1,1). For south output port evaluation, it considers nodes (2,0) and (1,1). This strategy works well with an odd-even routing function [1], as certain turns are eliminated for deadlock avoidance. However, with a fully adaptive routing function, its performance degrades due to limited knowledge.

3.2 Intra-region interference

Three RCA variants have been proposed: RCA-1D, -Fanin and -Quadrant [17]. RCA-1D transmits aggregated status information along each dimension. RCA-Fanin captures more congestion information by aggregating information from orthogonal dimensions as the status is transmitted. RCA-Quadrant improves accuracy over Fanin by separating congestion values for different network quadrants. No single RCA variant provides the best performance across all traffic patterns. Our experimental results show more interference and larger performance degradation for RCA-Quadrant and RCA-Fanin than for RCA-1D; therefore, we use RCA-1D as a baseline.

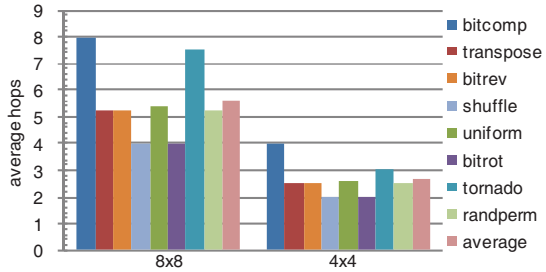


Fig. 3: The average hops for synthetic traffic.

Fig. 2(c) shows an intra-region[1] scenario for RCA-1D. All 16 nodes run the same application. When evaluating east output congestion, RCA-1D considers the status of nodes (0,1), (0,2) and (0,3). Similarly, it considers nodes (1,0), (2,0) and (3,0) when evaluating the south port. For destination node (2,2), the information from nodes (0,3) and (3,0) cause interference as they lie outside the minimum quadrant defined by (0,0) and (2,2); this packet will not traverse those nodes. This interference may result in poor output port selection and cause performance degradation. In other words, when evaluating output ports, RCA-1D considers the status of all nodes along each admissible direction and introduce excessive congestion information, which may degrade performance, especially considering traffic locality.

We compute the average hop count (AHP) to measure traffic locality for several synthetic traffic patterns [8] on 8×8 and 4×4 meshes. Most synthetic traffic has an AHP of less than 5.6 hops (average 5.58) and 3 hops (average 2.63) for the 8×8 and 4×4 mesh networks respectively as shown in Fig. 3. These patterns exhibit locality as most packets travel a short distance between source and destination. Thus, we need strategies to mitigate intra-application interference.

[1]We use region and application interchangeably.

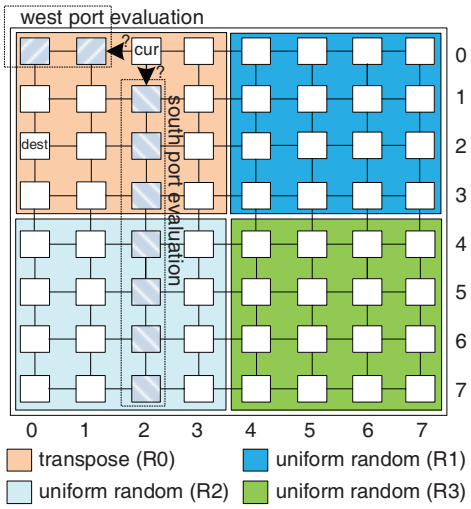


Fig. 4: An inter-region interference scenario for RCA-1D (the current router is (0,2) and the destination is (2,0)).

3.3 Inter-region interference

Fig. 4 illustrates a workload consolidation example for an 8×8 mesh network; similar scenarios will be prevalent in many-core systems. Here, there are 4 concurrent applications and each application is mapped to a 4×4 region. Region R0 is defined by nodes (0,0) and (3,3), R1 is defined by nodes (0,4) and (3,7), R2 is defined by nodes (4,0) and (7,3), and R3 is defined by nodes (4,4) and (7,7). Fig. 4 shows a packet whose current router is (0,2) and that needs to be routed to node (2,0). Even though traffic in R0 is isolated from traffic in other regions, RCA-1D considers the congestion status of nodes in R2 when selecting output ports for traffic belonging to R0. Obviously, this method introduces significant interference in output selection and reduces performance isolation.

To evaluate the effect of this inter-region interference, we assign transpose traffic to R0 and uniform random traffic to R1-R3. The performance of region R0 is presented in Fig. 5. For '*RCA-uni_region*' curve there is only one region (R0) in a 4×4 mesh network with transpose traffic; this latency reflects perfect isolation and no inter-region interference. The saturation throughput of RCA-1D is ∼65%. However, without isolation, the saturation throughput drops dramatically to ∼50% under workload consolidation, as shown with the '*RCA-multi_regions*(4%)' curve where R1, R2 and R3 all have a 4% injection rate (limited by their respective boundaries). For the '*RCA-multi_regions*(64%)' curve, R1 has a 64% injection rate while R2 and R3 remain at 4%; in this unbalanced scenario, the saturation throughput of R0 further decreases to 47% (see Section 5 for more detail). Clearly, the congestion information of R1, R2 and R3 greatly affects the routing selection in R0. RCA-1D couples the activities of otherwise independent applications, and this characteristic is not desirable for workload consolidation.

RCA could be extended to limit inter-region interference through statically configured cutoffs in boundary routers; this mechanism may be complex and boundaries would have to be computed off-line. As applications and their mappings may change during run-time, this mechanism has poor flexibility. More importantly, it cannot eliminate intra-region interference, which significantly affects performance for small regions as we will show in Section 5.

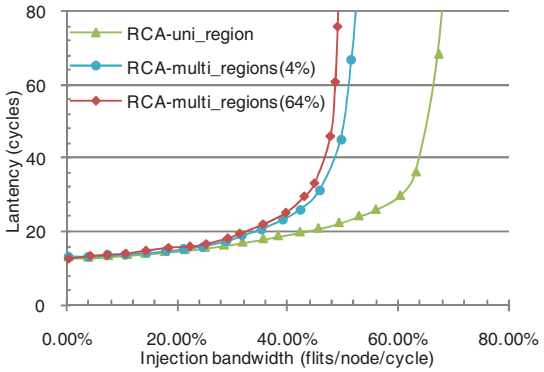


Fig. 5: Load-latency graph of region 0.

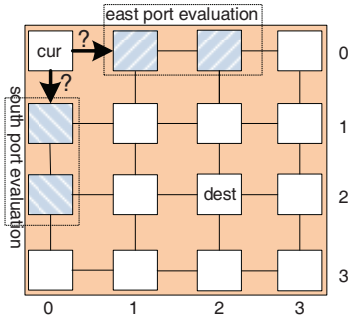


Fig. 6: A scenario for DBAR (the current router is (0,0) and the destination is (2,2)).

DBAR aims to reduce both intra- and inter-region interference by considering only the congestion of nodes in the minimum quadrant defined by the current and destination nodes. Fig. 6 shows a scenario similar to Fig. 2. When evaluating the east output, DBAR considers nodes (0,1) and (0,2); when evaluating the south output, DBAR considers nodes (1,0) and (2,0). This scheme leverages information from both neighboring and non-local nodes; it has more accurate knowledge about network congestion than LOCAL and NoP. DBAR does not consider congestion information from nodes (0,3) and (3,0) which eliminates interference. More importantly, for workload consolidation with each application mapped to a near convex region [5, 4], DBAR dynamically isolates routing for each region. In other words, the DBAR algorithm has neither intra- nor inter-region interference and provides sufficient adaptivity for congestion.

4. DESTINATION-BASED SELECTION STRATEGY DESIGN

The selection strategy in adaptive routing algorithms significantly impacts performance [1, 35, 14, 29]. An efficient selection strategy should ideally satisfy two goals: **high adaptivity** and **dynamic isolation** for workload consolidation. The selection strategy should leverage both local and non-local network congestion information for better accuracy. At the same time, it should not utilize excess information. More importantly, under workload consolidation, the selection strategy should offer dynamic isolation for diffe-

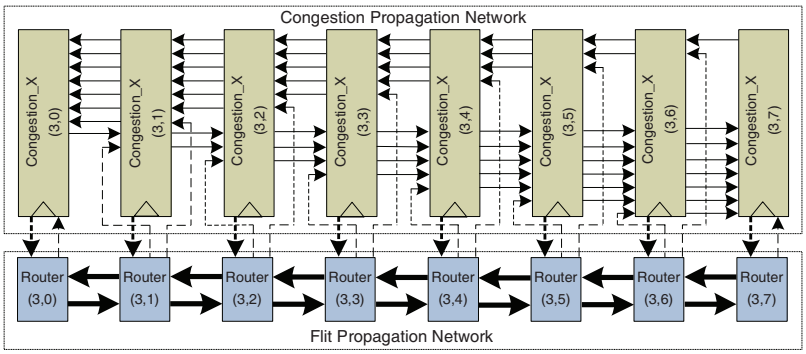


Fig. 7: The congestion information propagation network along one dimension. The bold arrows represent channels of multiple bits. The thin arrows represent channels of one bit. The dashed arrows are inside the router; we draw them to provide clarity for the congestion status propagation network.

rent applications, to avoid the negative effects of inter-region interference.

Before presenting the destination-based selection strategy, we explain the design of our low-cost congestion information propagation network. Each router forwards the number of available VCs to other routers in the same dimension. Each router has the congestion status of all other routers in the same dimension. We use the number of available VCs since it requires low wiring overhead for propagation. Other congestion metrics achieve similar performance.

DBAR selects the output port according to the weighted congestion value of each dimension; only nodes inside the quadrant defined by the current and destination nodes are considered in this weighted congestion calculation. By ignoring nodes residing outside the minimal quadrant, this congestion computation minimizes the intra-region interference and offers dynamic isolation between different regions.

4.1 Contention Information Propagation Network

In off-chip networks, bandwidth is constrained by pin limitations. However, NoCs can take advantage of abundant wiring to employ a dedicated network to exchange congestion information without adding traffic overhead [1, 17]. The dedicated congestion propagation network enables the router to leverage both local and non-local network information to accurately estimate network congestion. Both NoP and RCA utilize such a low-bandwidth monitoring network [1, 17]. NoP leverages this network to exchange free buffer slots of neighboring routers between adjacent routers. RCA leverages this network to obtain the congestion status of distant routers beyond those adjacent to neighboring routers. We focus on obtaining global information; the propagation network in RCA serves as the best comparison point.

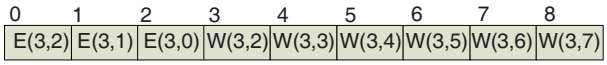
At each hop in RCA's congestion propagation network, the local status is aggregated with information from neighboring nodes and then propagated to upstream routers [17]. This implementation has two limitations. First, the aggregation logic combines local and distant information during transmission, making it impossible for the selection function to filter out superfluous information. Second, the aggregation logic introduces an additional cycle of latency per hop, leading to stale congestion information at distant routers. Based on these two observations, we propose a novel propagation network, which consumes only one cycle per tile, giving DBAR timelier congestion information. More importantly, the design makes it feasible for the selection function to filter out information based on the packet destination.

Fig. 7 shows the proposed congestion propagation network for the third row of an 8×8 mesh; the same structure is present in each row and column. Along a dimension, each router has a register (*congestion_X* or *congestion_Y*) to store the incoming congestion information. The incoming congestion information along with the local status are forwarded to the neighboring nodes in the next cycle via the congestion propagation channel.

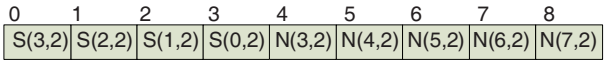
The width of each congestion propagation channel needs to be $\log(numVCs)$ to cover the range of free VCs. However, a coarser approximation of available VCs is sufficient to estimate congestion. For neighboring routers, making a fine distinction between available VCs will have little impact; for example, assume a packet can choose between two output ports with 5 and 6 available VCs respectively (8 total VCs). It is nearly equivalent to send the packet to either port since they are both lightly loaded. On the other hand, since the router weighs the incoming congestion information according to the distance from current router, it is also unnecessary to have accurate numbers for distant routers.

As we show in Section 7, one wire is sufficient for achieving high performance; the router forwards congestion information in an on/off manner. The threshold for indicating congestion (forwarding a 0) is 4 (out of 8 VCs); when 5 or more VCs are available, a 1 is forwarded to indicate no congestion. Coarse-grain congestion signals will toggle infrequently resulting in a low activity factor and low power consumption for this network.

Using this coarse-grain monitoring, both the *congestion_X* and *congestion_Y* registers are 9 bits wide. Incoming congestion information from routers in the same dimension is stored in 7 bits and other 2 bits store the conditions of two ports in the local router. The router weighs the incoming congestion information based on the distance from current router; the weight of incoming congestion information is halved for each additional hop. This ratio is chosen based on prior work [17] and practical implementation complexity. Adjacent bit positions of a register inherently maintain a step ratio of 0.5, thus we can easily implement this step ratio by putting the incoming congestion information in the appropriate positions in the registers.



(a) *Congestion_X* of Router(3,2).



(b) *Congestion_Y* of Router(3,2).

Fig. 8: The format of the congestion registers (E: East, W: West, N: North, S: South).

Fig. 8 shows the format of *congestion_X* and *congestion_Y* registers in router (3,2). Bit 0 of *congestion_X* stores the east input port status of current router. Bits 1 and 2 store the incoming congestion information from its nearest and one hop farther west neighbor: Routers (3,1) and (3,0), respectively. These first three bits are forwarded to the east neighbor: Router (3,3). Bit 3 of *congestion_X* stores the west input port status of current router, and the following five bits sequentially store the west input port status of the remaining east side routers based on distance. These six bits are forwarded to the west neighbor: Router (3,1).

Bits are stored in a similar fashion for *congestion_Y*. Bit 0 stores the south input port status of current router, followed by three bits storing the south input port status of the routers located to the north of current router. These four bits are forwarded to the south neighbor: Router (4,2). Bit 4 stores the north input port status of current router, and the following four bits stores the north input port status of the routers located to the south of current router. These five bits are forwarded to the north neighbor: Router (2,2).

RCA-Fanin and -Quadrant aggregate information from multiple dimensions. In their evaluation [17], this additional information can sometimes be helpful in selecting the output port. DBAR maintains information on a per-router basis for all routers in a single dimension. DBAR could be extended to incorporate additional information from fan-in routers; however, this modification would substantially increase the complexity of the congestion information propagation network.

4.2 DBAR Router Microarchitecture

Our DBAR router is based on a canonical VC router [8, 13]. The pipeline of the canonical VC router is composed of four stages: routing computation (RC), VC allocation (VA), switch allocation (SA) and switch traversal (ST). Link traversal (LT) requires one cycle to forward the flit to next hop. For high performance, the DBAR router applies speculative switch allocation [32]; VA and SA proceed in parallel at low network loads. We also leverage look-ahead adaptive routing computation to remove the RC stage from the critical path [24, 17, 15]; the router calculates at most two alternative output ports for the next hop. Advanced bundles [18, 25] encoding the packet destination ID traverse the link to the next hop while the flit is in the switch traversal stage as shown in Fig. 9.

Selection Metric Computation. The Selection Metric Computation (**SMC**) and Dimension Pre-selection (**DP**) modules are added to the router as shown in Fig. 10. The SMC module computes the dimension of the optimal output port for every possible destination using the congestion information stored in *congestion_X* and *congestion_Y*. An

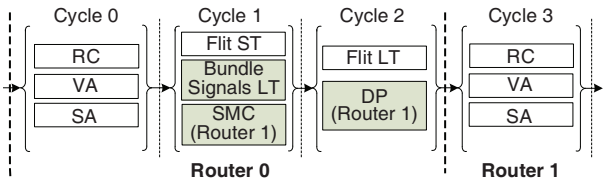


Fig. 9: The pipeline of DBAR router.

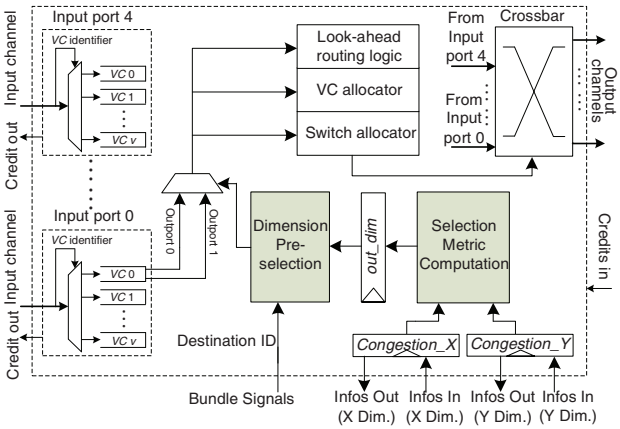


Fig. 10: DBAR router architecture.

additional register, *out_dim* stores the results of the **SMC**. With minimal routing, there are at most two admissible output ports (1 per dimension) for each destination. Due to these restrictions, the *out_dim* register uses one bit to represent the optimal output port for each destination. If the value is '0', the optimal output port is along dimension X; otherwise, it is along dimension Y.

Fig. 11 illustrates the pseudo-code of **SMC** module to compute the optimal output dimension for a packet whose destination is the pos^{th} bit position of the *out_dim* register. Packets forwarded to the local node are excluded from this logic. Along each dimension, only those bit positions in the *congestion_X* and *congestion_Y* registers storing congestion information for nodes inside the quadrant defined by the current and the pos^{th} nodes are chosen. The chosen values are the congestion status metric for each dimension. According to the relative magnitude of the congestion status for the X and Y dimensions, the **SMC** sets the value of the pos^{th} bit in the *out_dim* register. If their magnitudes are equal, DBAR randomly chooses an output dimension. Since the **SMC** module only examines bit positions representing those nodes inside the quadrant defined by current node and the pos^{th} node along each dimension, interference is not introduced. At the same time, DBAR utilizes the information from non-local routers to improve its ability to avoid congestion.

Dimension Pre-selection. To remove the output port selection procedure from the critical path of the DBAR router, the Dimension Pre-selection (**DP**) module (shown in Fig. 12) accesses the *out_dim* register one cycle ahead of the flit's arrival. The value of *out_dim* is computed out by **SMC** module in the previous cycle (see Fig. 9).

The **DP** module selects the corresponding bit position of the *out_dim* register according to the destination encoded in an advanced bundle. Six *XOR* gates and a *NOR* gate are

```
1:  if ( pos_x < cur_x )
2:      tmp_x[0:cur_x-pos_x-1] ← congestion_X[1:cur_x-pos_x];
3:  else if (pos_x > cur_x )
4:      tmp_x[0:pos_x-cur_x-1] ← congestion_X[cur_x+2:pos_x+1];
5:  else {
6:      out_dim[pos] ← 1;
7:      return;}
8:  if ( pos_y < cur_y )
9:     tmp_y[0:cur_y-pos_y-1] ← congestion_Y[1:cur_y-pos_y];
10: else if (pos_y > cur_y )
11:     tmp_y[0:pos_y-cur_y-1] ← congestion_Y[cur_y+2:pos_y+1];
12: else {
13:     out_dim[pos] ← 0;
14:     return;}
15: if( tmp_x < tmp_y )
16:     out_dim[pos] ←1;
17: else if( tmp_x > tmp_y )
18:     out_dim[pos] ← 0;
19: return;
```

Fig. 11: The pseudo-code of SMC module. (cur_y, cur_x) and (pos_y, pos_x) are the positions of current and pos^{th} router respectively. The initial value of tmp_x and tmp_y are 7-bit 0s.

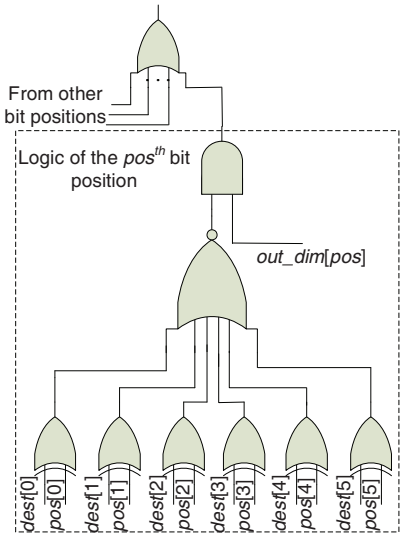


Fig. 12: Hardware implementation of DP module.

used to generate the mask signal. If the destination ID is equal to *pos*, the mask signal is set to '1'; otherwise, it is set to '0'. An *AND* gate masks or unmasks the pos^{th} bit value of the *out_dim* register. Finally, the logic values of all bit positions are combined by an *OR* gate to generate the port selection signal. This *OR* gate combines the values of the other bit positions in the *out_dim* register that do not correspond to packet's destination and have been masked off. The head flit encodes the admissible output ports computed in last hop. When the head flit arrives current node, it chooses the output port according to the result of the **DP** module. Using the logical effort model [32], the delay of the **DP** module is ∼8.1 FO4. If the **DP** module were added to the VA stage, the critical path would increase from 20 FO4 to 28.1 FO4. Advanced bundles serve to avoid this increase.

5. EVALUATION

We modified the Booksim simulator [8] to model the microarchitecture and pipeline discussed in Section 4. The router pipeline is two cycles plus one cycle for link traversal. DOR is chosen for the deterministic routing algorithm. We implement a locally adaptive routing algorithm (LOCAL), NoP and RCA-1D[2]. To be fair, DBAR, RCA, NoP and LOCAL all employ a fully adaptive routing function based on Duato's theory [11].

We use 8 VCs with 5 flit buffers each. We use 4×4 and 8×8 mesh topologies. The packet length is uniformly distributed between 1 and 6 flits. The simulator is warmed up for 10,000 cycles and then the average performance is measured over another 100,000 cycles. Both synthetic traffic patterns [8] and application traces from scientific [39] and commercial workloads [37, 38] are used. Application traces are obtained from a full system simulator configured as shown in Tab. 1. Workload details are presented in Tab. 2.

Tab. 1: Full system simulation configuration.

Cores	16 in-order cores
Memory system	
L1 I/D Cache	32 KB 2-way set associative (1 cycle latency)
Private L2 Cache	512 KB 4-way set associative (6 cycles latency)
Shared L3 Cache	16 MB 16 way set associative (12 cycles latency)
Main Memory Latency	100 cycles

Tab. 2: Benchmark description.

Benchmark	Description
Barnes	8K particles, full end-to-end run including initialization
Ocean	512×512 full end-to-end run (parallel phase only)
Radiosity	-room -batch -ae 5000 -en 0.050 -b 0.10 (parallel phase only)
Raytrace	car input (parallel phase only)
SPECjbb	Standard java server workload utilizing 24 warehouses, executing 200 requests
SPECweb	Zeus Web Server 3.3.7 servicing 300 HTTP requests
TPC-H	Transaction Processing Council's Decision Support System Benchmark, using IBM DB2 v6.1, running query 12 with a 512MB database and 1GB of memory
TPC-W	Transaction Processing Council's Web e-commerce benchmark, DB Tier, browsing mix, 40 transactions

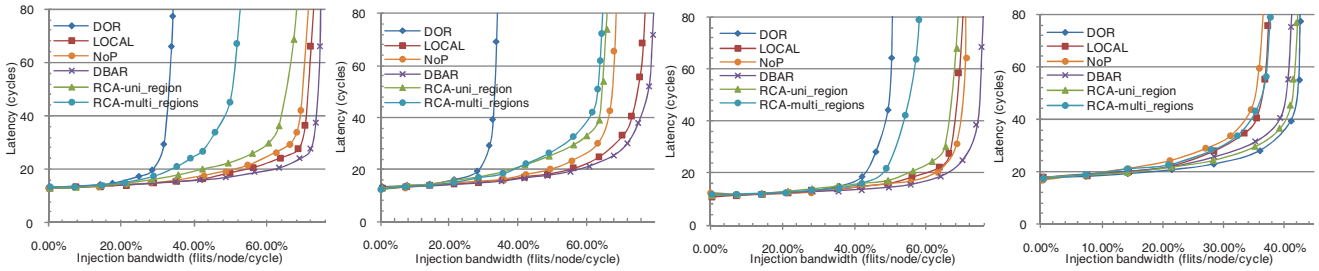
5.1 Single Region Performance

To highlight the impact of insufficient congestion information for LOCAL and NoP and the intra-region interference for RCA, we evaluate the performance of these five algorithms in two single application configurations: 4×4 and 8×8 mesh networks. There is only one traffic pattern throughout the whole network.

Synthetic Traffic Results. Fig. 13 and Fig. 14 give the latency results for the five algorithms using transpose, bit reverse, shuffle and bit complement traffic patterns in 4×4 and 8×8 mesh networks, respectively.

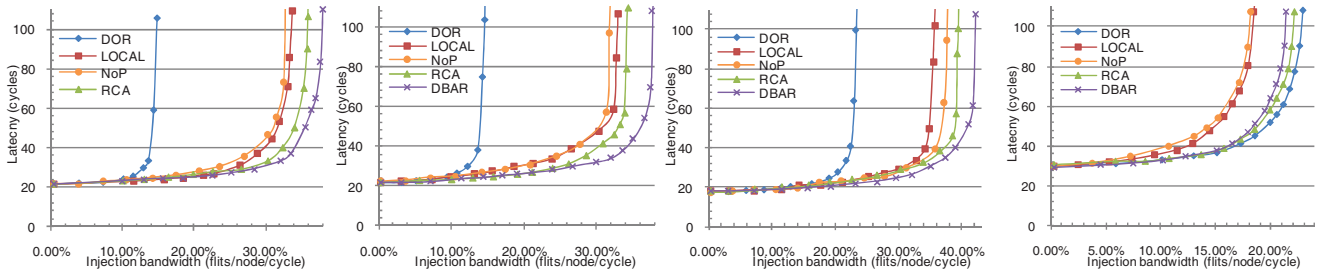
In the 4×4 mesh network, DBAR has the best performance on these four traffic patterns as RCA suffers from intra-region interference. There is one exception: for bit complement, RCA's saturation point is 2.1% higher[3]. Bit complement has the largest AHP with 4 hops in a 4×4

[2]RCA-1D is referred to as RCA throughout the evaluation.
[3]The saturation point is the point at which the average latency is 3 times the zero load latency.



(a) Transpose. (b) Bit reverse. (c) Shuffle. (d) Bit complement.

Fig. 13: Routing algorithm performance for a 4×4 mesh network (region). *RCA-uni_region* and *RCA-multi_region* give the performance of RCA for a single region and multiple regions respectively.



(a) Transpose. (b) Bit reverse. (c) Shuffle. (d) Bit complement.

Fig. 14: Routing algorithm performance for an 8×8 mesh network with a single region.

network (Fig. 3); this AHP mitigates the intra-region interference. LOCAL and NoP perform the worst for bit complement traffic due to their limited knowledge. The small AHP (2.5 hops) of transpose traffic leads to RCA performing the worst among all four adaptive algorithms. DBAR, LOCAL and NoP offer similar performance for transpose traffic with ~13% improvement in saturation throughput versus RCA. DBAR has a significant improvement of 21.9% relative to RCA for bit reverse.

DBAR shows 10.2% and 8.5% saturation throughput improvement over LOCAL for shuffle and bit complement traffic. These patterns cause global congestion and the shortsightedness of the locally adaptive strategy makes it unable to avoid congested areas. The saturation throughput improvements of DBAR against NoP are 17.7% and 11.1% for bit reverse and bit complement traffic respectively. NoP overlooks the status of neighboring nodes. Comparing the performance of LOCAL against NoP further illuminates this limitation. This phenomenon validates our weighting mechanism placing more emphasis on closer nodes.

LOCAL outperforms RCA on a 4×4 mesh; intra-region interference leads RCA to make inferior selection decisions. However, in the 8×8 mesh, DBAR and RCA offer the best performance, while LOCAL has inferior performance. RCA's improvement comes from the weighted mechanism in the congestion propagation network. The weight of the congestion information halves for each hop; the effect of intra-region interference from distant nodes diminishes. This interference reduction is a result of the high AHP of 5.58 for these patterns. However, the AHP on the 4×4 network is 2.63, which is not large enough to hide the negative effect of interference.

Although the weighted aggregation mechanism mitigates some interference in the 8×8 mesh network, DBAR still outperforms RCA by 11.1% for bit reverse traffic. Compared with the 4×4 network, DBAR further improves the saturation throughput for shuffle and bit complement versus LOCAL by 12.4% and 16.5%. The shortsightedness of LOCAL has a stronger impact in a larger network. Similar trends are seen for NoP. For most traffic, DOR's rigidity prevents it from avoiding congestion.

Application Results. Fig. 15 shows average packet latencies normalized to DOR in a 4×4 network for several scientific and commercial applications. Since *Barnes* exhibits global load balance and a low injection rate, DOR offers the best performance. For the other applications, DBAR has the lowest latency. For most applications including *Ocean*, *SPECjbb*, *TPC-H* and *TPC-W*, RCA has the worst latency; this is consistent with the synthetic results. NoP has larger latency than LOCAL for most applications; its ignorance of neighboring nodes results in sub-optimal selections. *Raytrace* and *TPC-H* have the largest latency reductions of 25.2% and 19.8% for DBAR versus LOCAL. These two applications have high injection rates, thus favoring the algorithm with higher throughput. The average latency reduction is 10.8% for DBAR versus LOCAL.

5.2 Multiple Region Performance

We evaluate three multiple-region configurations: two regular (small and medium sizes shown in Fig. 4 and Fig. 16) and one irregular region configurations (Fig. 17). In all configurations, we focus on the performance of R0.

Small-Sized Regular Region Results. In the first and second configurations (Fig. 4 and Fig. 16), regions R1, R2 and R3 (in Fig. 4 only) run uniform traffic with 4% injection rates while we vary the pattern in R0. For the regular region configuration, LOCAL, NoP and DBAR do not have inter-region interference, since they only consider the congestion status of nodes belonging to the same region when

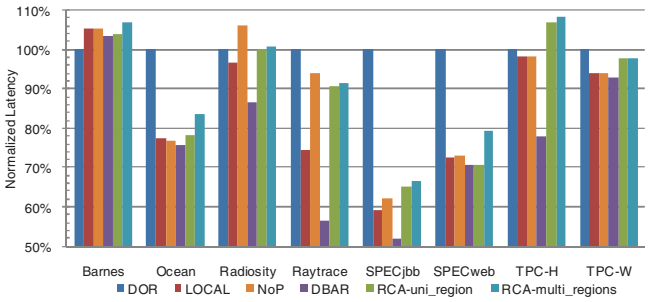


Fig. 15: Performance for application traces.

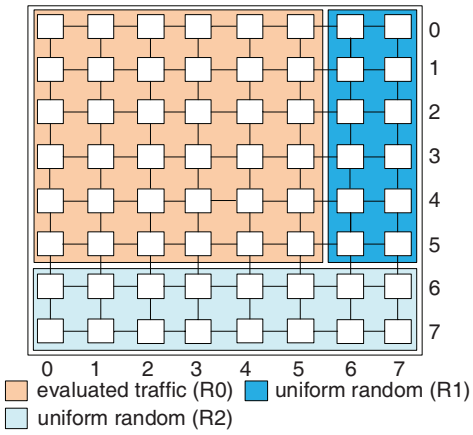


Fig. 16: Medium-sized region configuration for an 8×8 mesh network.

making selections. Thus, for the first region configuration (Fig. 4), these 3 algorithms and DOR have the same performance as shown in Fig. 13. However, RCA's performance suffers from inter-region interference, since it considers the congestion status of all nodes along each dimension when selecting the output port. The '*RCA-multi_regions*' curves in Fig. 13 show RCA's performance for the multiple regions configuration.

Compared with the single region, RCA's performance declines; RCA suffers not only from intra-region interference, but also from inter-region interference. Transpose and shuffle see 22.7% and 16.9% drops in saturation throughput. For bit reverse traffic, the performance degradation is minor; the intra-region interference has already significantly degraded RCA's performance and hides the effect of inter-region interference. DBAR maintains its performance for this configuration, thus revealing a clear advantage. The average saturation throughput improvement is 25.2% with the maximum improvement of 46.1% for transpose traffic. Fig. 15 ('*RCA-multi_regions*') shows that RCA's latency increases for all applications compared to the single region configuration (in multiple regions configuration, R1-R3 run uniform random traffic with 4% injection rates). *SPECweb* has the maximum latency increase of 11.1%.

Routers at the boundary of R1 and R2 strongly affect R0's performance, since some of their input ports are never used. For example, the west input VCs of router (0,4) are always available since no packets arrive at this router from the west. The interference from internal nodes of R1 and R2 is partially masked by RCA's weighting mechanism at

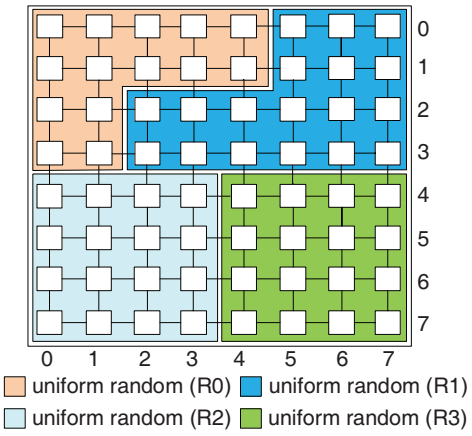


Fig. 17: Irregular region configuration for an 8×8 mesh network.

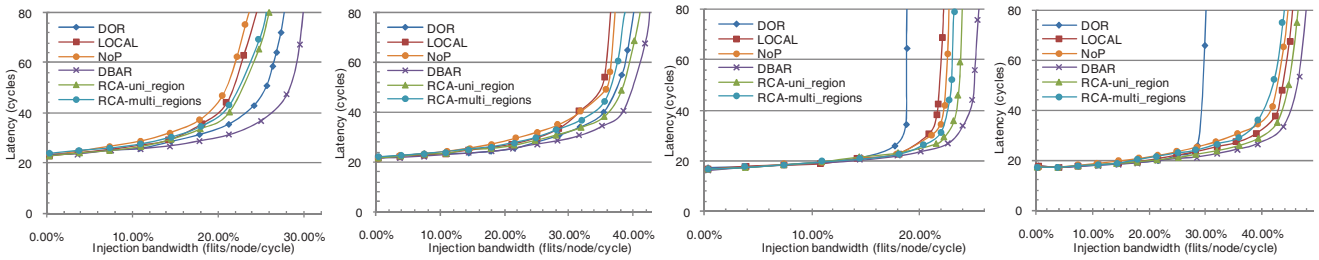
these boundary nodes with 8 free VCs. This explains why R0's saturation point only decreases from 50% to 47% when the injection rate of R1 increases from 4% to 64% in Fig. 5.

Medium-Sized Regular Region Results. Fig. 18 shows the performance of the algorithms in the second multiple-region configuration (Fig. 16). This configuration reveals the relationship between region size and performance. We use four traffic patterns: matrix transpose [20], tornado, hot spot and random permutation. Under hot spot traffic, three hot spot nodes receive an extra portion (20%) of traffic in addition to the regular uniform traffic. Such hot spots may occur when a disproportionate amount of traffic travels to memory controllers. Random permutation is the average performance of 1000 permutations from total 36! possible permutations [36].

DBAR provides the highest performance for all patterns. With the increase in region size, RCA has better performance relative to LOCAL and NoP, which is consistent with the trend revealed in the single region configuration evaluation. For medium-sized region, the shortsightedness of LOCAL and NoP begin to limit their performance, as compared with the performance of the 4×4 mesh network. Although RCA still suffers from inter-region interference, the saturation throughput drop is not as dramatic as Fig. 13 shows. The maximum throughout drop is 7.6% for random permutation traffic. Larger AHP and the weighted mechanism help to mitigate the inter-region interference.

Irregular Region Results. Fig. 17 shows non-rectangular regions. The isolation boundaries of R0 and R1 are the minimal rectangle surrounding these regions; some nodes receive traffic from both regions. We show the performance of R0 while varying the injection rate of R1 from low load (4%) to high load (55%) in Fig. 19; the injection rates of R2 and R3 are fixed at 4%. Uniform random traffic is run in all regions.

For both high and low loads in R1, DBAR has the best performance. As the load in R1 increases, the performance of all algorithms declines. For low load in R1, RCA has the second highest saturation throughput. Two rows of R0 have 5 routers; LOCAL and NoP are not sufficient to avoid congestion. When R1 has a high injection rate, the saturation points decline for DOR, LOCAL, NoP, RCA and DBAR by 7%, 7%, 6.8%, 6.7% and 4% respectively; DBAR



(a) Matrix-transpose. (b) Tornado. (c) Hot spots. (d) Random permutation.

Fig. 18: Routing algorithm performance for a 6×6 mesh network (region). *RCA-uni_region* and *RCA-multi_region* give the performance of RCA for a single region and multiple regions respectively.

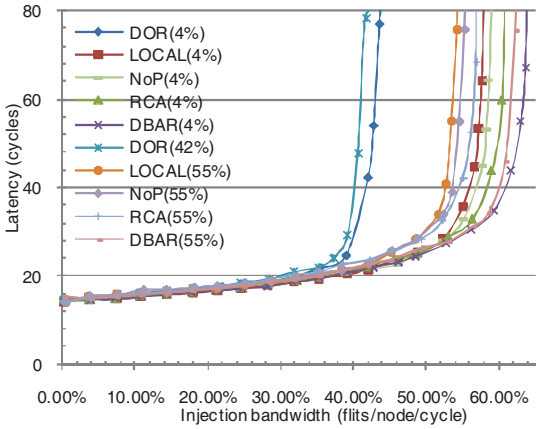


Fig. 19: Performance of R0 with irregular regions.

Tab. 3: Average saturation throughput improvement of DBAR.

network	LOCAL	NoP	RCA	RCA_multi
4×4	7.2%	8.8%	10.4%	25.2%
6×6	13.5%	11.7%	9.6%	14.1%
8×8	12.6%	14.9%	4.7%	-
irregular	16.5%	14.3%	-	6.8%

shows the least performance degradation since it offers the best isolation between these two regions. DBAR can provide more predictable performance when running multiple applications.

Summary. In a workload consolidation scenario, different concurrent applications will be mapped to different region sizes (ranging from small to large) according to their intrinsic parallelism. However, with small regions, RCA suffers from intra- and inter-region interference, while LOCAL and NoP are limited by shortsightedness for medium and large-sized regions. Neither of these algorithms provide good performance for workload consolidation on a many-core platform.

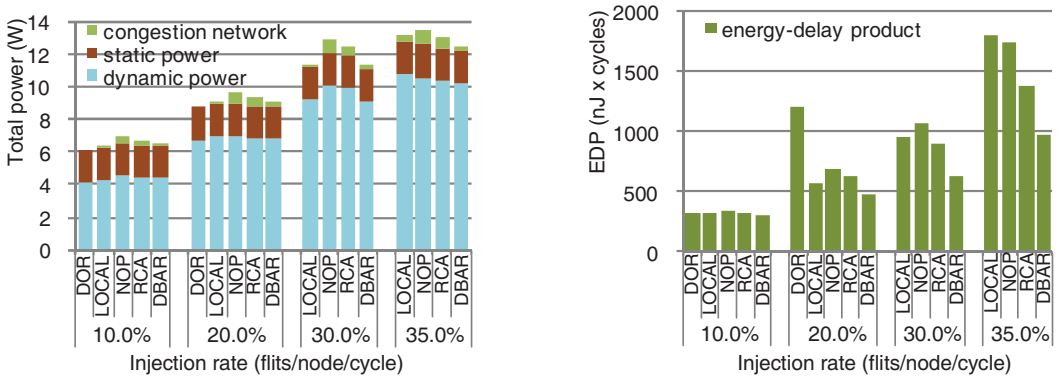
Tab. 3 lists the average saturation throughput improvement of DBAR against other algorithms for different configurations. DBAR provides better performance than the best baseline for all evaluated configurations and it shows the smallest performance degradation with multiple irregular regions. Thus, DBAR is well suited to workload consolidation.

6. OVERHEAD: WIRING AND POWER CONSUMPTION

Wiring Overhead. DBAR, RCA, NoP and LOCAL all require some wiring overhead to transmit congestion information. DBAR introduces 8 additional wires for each dimension. RCA's congestion network uses 8 wires in each direction for a total of 16 per dimension. Although RCA can be optimized to transmit congestion status in a bit-serial manner using only a single wire per direction, we do not consider this design. NoP requires $4 \times \log(numVCs) = 12$ wires per direction; there are 24 wires for one dimension. LOCAL requires $\log(numVCs) = 3$ wires in each direction for 6 total wires per dimension. Given a state-of-the-art NoC design with 128-bits channels [28], the overhead of DBAR is just 3.125% versus 6.25%, 9.375% and 2.34% for RCA, NoP and LOCAL, respectively. DBAR has a modest overhead; abundant wiring on chip is able to accommodate these wires.

Power Consumption. We leverage an existing NoC power model [30], which divides the total power consumption into three main components: channels, input buffers and router control logic including crossbar traversal, crossbar control and output control module. Leakage power is included for buffers and channels. We also model the power consumption of the congestion propagation network and the additional modules of DBAR. The activity of these components is obtained from a cycle-accurate simulator. We use a 32nm technology process with a 1 GHz clock frequency. The process parameters are obtained from ITRS roadmap [22].

Fig. 20(a) illustrates the average power for transpose traffic with different injection rates. Since DOR cannot support injection rates higher than 20%, there are no results for 30% and 35% injection rates. The increased hardware complexity, especially the congestion propagation network of adaptive routing algorithms results in a higher average power than the simple DOR algorithm. Comparing these four adaptive routers, LOCAL and DBAR have the lowest power since they have the lowest wiring overhead. NoP has the highest power. LOCAL need 6 additional wires, which is less than DBAR, but these wires have a higher activity factor than DBAR. For a 20% injection rate, the activity of DBAR's congestion propagation network is 15.8% versus 17.5% of LOCAL. This smaller activity factor mitigates the increased power of DBAR's congestion propagation network. For a 35% injection rate, LOCAL consumes more power than DBAR. The adaptive routing algorithm accelerates packet transmission, showing a significant energy-delay



(a) Power. (b) EDP.

Fig. 20: Power consumption results for transpose traffic.

product (EDP) advantage. As shown in Fig. 20(b), DBAR provides smallest energy-delay product for medium (20%) and high injection rates (30% and 35%).

7. DBAR DESIGN SPACE EXPLORATION

Number of Propagation Wires. Tab. 4 lists the saturation throughput of DBAR with 1-, 2- and 3-bit wide propagation networks for two network sizes. Wider propagation networks can transmit VC utilization with finer granularity. The increase in wiring brings only minor performance improvements, and these performance gains decrease as the network scales. The trend is similar with 16 VCs per port. The performance gain with a 4-bit propagation network is marginal compared to 1 bit. When the number of VCs is larger than 8, crossbar conflicts are more limiting than head of line conflicts [23]. Making a fine distinction about the available VCs has little practical impact. By using 1-bit wire for congestion information propagation, we introduce only 8 additional wires along each dimension.

DBAR Scalability. The cost of scaling DBAR to a larger network increases linearly as N 1-bit congestion propagation wires are needed for an $N \times N$ network. For a 16×16 network, this represents a 6.25% overhead with 128 bit channels. The size of the added registers in DBAR also increase linearly. The latency of the **DP** module increases logarithmically with network radix; however this delay is not on the critical path so it will not increase the router cycle time. DBAR is a cost effective solution for many-core networks.

Congestion Propagation Delay. In addition to eliminating interference, our novel congestion network operates with only a 1 cycle per hop delay compared to 2 cycles per hop in RCA. To isolate this effect from interference effects, we compare DBAR with a 1 cycle per hop and a 2 cycle per hop congestion propagation network. The timeliness of the 1 cycle per hop network improves saturation throughput by up to 5% over the 2-cycle design (for shuffle traffic pattern).

8. CONCLUSIONS

Current routing algorithms cannot provide high performance for workload consolidation. The shortsightedness of locally adaptive routing algorithms limits their performance for medium and large-sized networks, while globally adaptive routing algorithms suffer from both intra- and inter-region interference for multiple regions. Interference across regions can occur even if packets of a given region never traverse nodes of another region; the interference comes from propagating congestion information across region boundaries. By leveraging a novel congestion information propagation network, the proposed DBAR algorithm provides both high adaptivity for network congestion and dynamic isolation to eliminate interference. Experimental results show that DBAR can offer better performance for small, medium and large-sized networks. The wiring overhead of DBAR is only 3.125%. DBAR provides the lowest energy-delay product for medium and high loads. DBAR is topology-agnostic; future work will extend DBAR to additional topologies beyond mesh networks. We also plan to explore fault tolerance in the context of DBAR.

Acknowledgments

We thank the anonymous reviewers for their comments and suggestions for improving this work. We also thank Mingche Lai, Libo Huang, and Wei Shi of NUDT and Robert Hesse and Sam Vafaee of UofT for their valuable feedback. We thank the CVA group of Stanford University for sharing booksim and especially thank Daniel Becker for his help.

This work is supported by the University of Toronto, the Natural Sciences and Engineering Research Council (NSERC) of Canada, 973 Program of China (2007CB310901), NSFC (61070037, 60873015, 60736013, 61025009, 60903039), Education Foundation of China (20094307120012), NUDT Innovation Foundation For Excellent Postgraduate (B090602) and Hunan Provincial Innovation Foundation For Postgraduate (CX2010B032).

9. REFERENCES

[1] G. Ascia, V. Catania, M. Palesi, and D. Patti. Implementation and analysis of a new selection strategy for adaptive routing in networks-on-chip. *Computers, IEEE Transactions on*, 57(6):809 –820, June 2008.

[2] S. Bell et al. TILE64 - processor: A 64-core SoC with mesh interconnect. In *ISSCC 2008*, pages 88 –598, February 2008.

[3] G.-M. Chiu. The odd-even turn model for adaptive routing. *Parallel and Distributed Systems, IEEE Transactions on*, 11(7):729 –738, July 2000.

[4] C.-L. Chou and R. Marculescu. Run-time task allocation considering user behavior in embedded multiprocessor

Tab. 4: The saturation points of DBAR using different width propagation networks.

		bitcomp	transpose	bitrev	shuffle	uniform	bitrot	tornado	randperm	average improvement
(4 × 4)	1-bit	39.6%	74.0%	78.5%	74.8%	71.2%	81.2%	72.8%	52.9%	-
	2-bits	41.2%	74.5%	78.6%	76.8%	71.3%	82.4%	72.9%	53.1%	1.33%
	3-bits	42.0%	74.8%	78.7%	77.4%	71.3%	82.6%	73.4%	53.1%	1.92%
(8 × 8)	1-bit	21.2%	35.4%	36.0%	40.8%	35.6%	43.2%	25.2%	28.6%	-
	2-bits	21.4%	36.4%	38.5%	40.9%	36.4%	43.3%	25.5%	28.6%	1.12%
	3-bits	21.5%	36.6%	38.7%	41.6%	36.6%	43.4%	25.7%	28.6%	1.72%

networks-on-chip. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 29(1):78 –91, January 2010.

[5] C.-L. Chou, U. Ogras, and R. Marculescu. Energy- and performance-aware incremental mapping for networks on chip with multiple voltage levels. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 27(10):1866 –1879, October 2008.

[6] W. Dally and C. Seitz. Deadlock-free message routing in multiprocessor interconnection networks. *Computers, IEEE Transactions on*, C-36(5):547 –553, May 1987.

[7] W. Dally and B. Towles. Route packets, not wires: on-chip interconnection networks. In *DAC 2001*, pages 684 – 689, May 2001.

[8] W. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[9] W. J. Dally and H. Aoki. Deadlock-free adaptive routing in multicomputer networks using virtual channels. *Parallel and Distributed Systems, IEEE Transactions on*, 4:466–475, April 1993.

[10] J. Duato. A new theory of deadlock-free adaptive routing in wormhole networks. *Parallel and Distributed Systems, IEEE Transactions on*, 4(12):1320 –1331, December 1993.

[11] J. Duato. A necessary and sufficient condition for deadlock-free adaptive routing in wormhole networks. *Parallel and Distributed Systems, IEEE Transactions on*, 6(10):1055 –1067, October 1995.

[12] J. Duato. A necessary and sufficient condition for deadlock-free routing in cut-through and store-and-forward networks. *Parallel and Distributed Systems, IEEE Transactions on*, 7(8):841 –854, August 1996.

[13] N. Enright Jerger and L. Peh. *On-Chip Networks*. Morgan and Claypool Publishers, San Francisco, CA, USA, 1 edition, 2009.

[14] W.-C. Feng and K. G. Shin. Impact of selection functions on routing algorithm performance in multicomputer networks. In *ICS 1997*, pages 132–139, July 1997.

[15] M. Galles. Spider: a high-speed network interconnect. *Micro, IEEE*, 17(1):34 –39, January-February 1997.

[16] C. Glass and L. Ni. The turn model for adaptive routing. In *ISCA 1992*, pages 278 –287, June 1992.

[17] P. Gratz, B. Grot, and S. Keckler. Regional congestion awareness for load balance in networks-on-chip. In *HPCA 2008*, pages 203 –214, February 2008.

[18] P. Gratz, K. Sankaralingam, H. Hanson, P. Shivakumar, R. McDonald, S. Keckler, and D. Burger. Implementation and evaluation of a dynamically routed processor operand network. In *NOCS 2007*, pages 7 –17, May 2007.

[19] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar. A 5-GHz mesh interconnect for a Teraflops processor. *Micro, IEEE*, 27(5):51 –61, September-October 2007.

[20] J. Hu and R. Marculescu. DyAD - smart routing for networks-on-chip. In *DAC 2004*, pages 260 – 263, June 2004.

[21] J. Hu and R. Marculescu. Energy- and performance-aware mapping for regular NoC architectures. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 24(4):551 – 562, April 2005.

[22] ITRS. International Technology Roadmap for Semiconductors, 2007 edition. `http://www.itrs.net`, 2007.

[23] M. Karol, M. Hluchyj, and S. Morgan. Input versus output queueing on a space-division packet switch. *Communications, IEEE Transactions on*, 35(12):1347 – 1356, December 1987.

[24] J. Kim, D. Park, T. Theocharides, N. Vijaykrishnan, and C. Das. A low latency router supporting adaptivity for on-chip interconnects. In *DAC 2005*, pages 559 – 564, June 2005.

[25] A. Kumar, P. Kundu, A. Singh, L.-S. Peh, and N. Jha. A 4.6Tbits/s 3.6GHz single-cycle NoC router with a novel switch allocator in 65nm CMOS. In *ICCD 2007*, pages 63 –70, October 2007.

[26] T. Lei and S. Kumar. A two-step genetic algorithm for mapping task graphs to a network on chip architecture. In *DSD 2003*, pages 180 – 187, September 2003.

[27] M. Li, Q.-A. Zeng, and W.-B. Jone. DyXY - a proximity congestion-aware deadlock-free dynamic routing method for network on chip. In *DAC 2006*, pages 849 –852, June 2006.

[28] D. llitzky, J. Hoffman, A. Chun, and B. Esparza. Architecture of the scalable communications core's network on chip. *Micro, IEEE*, 27(5):62 –74, September-October 2007.

[29] J. C. Martínez, F. Silla, P. López, and J. Duato. On the influence of the selection function on the performance of networks of workstations. In *ISHPC 2000*, pages 292–299, October 2000.

[30] G. Michelogiannakis, D. Sanchez, W. Dally, and C. Kozyrakis. Evaluating bufferless flow control for on-chip networks. In *NOCS 2010*, pages 9 –16, May 2010.

[31] O. Mutlu and T. Moscibroda. Parallelism-aware batch scheduling: Enhancing both performance and fairness of shared DRAM systems. In *ISCA 2008*, pages 63 –74, June 2008.

[32] L.-S. Peh and W. Dally. A delay model and speculative architecture for pipelined routers. In *HPCA 2001*, pages 255 –266, May 2001.

[33] R. S. Ramanujam and B. Lin. Destination-based adaptive routing on 2D mesh networks. In *ANCS 2010*, pages 19:1–19:12, October 25-26 2010.

[34] S. Rodrigo, J. Flich, J. Duato, and M. Hummel. Efficient unicast and multicast support for CMPs. In *MICRO 2008*, pages 364 –375, November 2008.

[35] L. Schwiebert and R. Bell. Performance tuning of adaptive wormhole routing through selection function choice. *J. Parallel Distrib. Comput.*, 62:1121–1141, July 2002.

[36] A. Singh, W. Dally, A. Gupta, and B. Towles. GOAL: a load-balanced adaptive routing algorithm for torus networks. In *ISCA 2003*, pages 194 – 205, June 2003.

[37] SPEC. SPEC benchmarks. `http://www.spec.org`, 2009.

[38] TPC. TPC benchmarks. `http://www.tpc.org`, 2008.

[39] S. Woo, M. Ohara, E. Torrie, J. Singh, and A. Gupta. The SPLASH-2 programs: characterization and methodological considerations. In *ISCA 1995*, pages 24 – 36, June 1995.

[40] S. Zhuravlev, S. Blagodurov, and A. Fedorova. Addressing shared resource contention in multicore processors via scheduling. In *ASPLOS 2010*, pages 129–142, March 2010.